高等院校电子信息类教材

数字逻辑电路

魏达 高强 金玉善 曹英晖 编著

科学出版社
www.sciencep.com

数字逻辑电路

高等院校电子信息类教材

数字逻辑电路

魏 达 高 强 金玉善 曹英晖 编著

科 学 出 版 社

北 京

内 容 简 介

本书重点介绍了数字逻辑电路和数字系统的基础理论和方法，系统地阐述了以下内容：数制与编码、逻辑代数基础、逻辑门电路、组合逻辑电路的分析与设计、时序逻辑电路的分析与设计、半导体存储器和可编程逻辑器件、数字系统的分析与设计以及硬件描述语言。

本书可作为计算机、电子、通信及自动化等专业的本科生教材，也可供相关领域的工程技术人员参考。

图书在版编目(CIP)数据

数字逻辑电路/魏达等编著. —北京：科学出版社，2005
(高等院校电子信息类教材)
ISBN 7-03-014622-0

Ⅰ.数… Ⅱ.魏… Ⅲ.数字电路：逻辑电路-高等学校-教材
Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2004)第 134202 号

责任编辑：马长芳 资丽芳/责任校对：张 琪
责任印制：钱玉芬/封面设计：陈 敬

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com
源海印刷有限责任公司印刷
科学出版社发行 各地新华书店经销
*
2005年1月第 一 版 开本：B5 (720×1000)
2005年1月第一次印刷 印张：21 3/4
印数：1—4 000 字数：425 000

定价：28.00 元

(如有印装质量问题，我社负责调换〈环伟〉)

前　言

数字逻辑电路是计算机科学与技术(类)大学本科专业重要的专业基础核心课程之一，主要研究数字电路与逻辑设计的理论与方法。

数字技术是当前发展最快的学科之一，现代数字系统设计技术是计算机专业学生学习的重要内容。本教材重点介绍了数字逻辑电路的基本概念与基本方法。数字系统作为一个整体的系统，涉及了数字系统技术许多方面的知识，因此，从系统的角度和观点来进行本教材的编写，使学生获得数字电路与逻辑设计方面的基本理论、基本知识和基本技能，培养学生分析问题和解决问题的能力，使学生真正从数字系统技术体系的高度把握各种数字系统的设计技术是本教材的编写目的所在。

数字逻辑电路是“计算机组成原理”、“计算机体系结构”、“微型计算机系统”、“计算机接口技术”等课程的基础，对理解计算机的工作原理有十分重要的作用。它的主要内容包括数制与编码、逻辑代数基础、逻辑门电路、组合逻辑电路、时序逻辑电路、编程逻辑、ISP 技术、数字系统等。本书共 10 章：第一章和第二章主要介绍了数制、数制的转换、BCD 编码、可靠性编码、带符号数的表示、逻辑代数基础、最小项最大项性质、卡诺图化简法及列表化简法；第三章主要介绍了 TTL 门电路及 CMOS 门电路的工作原理和外部特性；第四章主要介绍了组合逻辑电路分析与设计的基本方法及各种常用组合逻辑电路的原理与应用；第五章至第七章主要介绍了触发器的基本工作原理及功能、同步时序逻辑电路分析与设计方法、常用时序逻辑电路的工作原理及应用、脉冲型异步时序逻辑电路的分析与设计方法、电平型异步时序逻辑电路的分析与设计的方法及电平型异步时序逻辑电路的竞争与冒险；第八章主要介绍了半导体存储器和可编程逻辑器件的基本概念与基本原理、PLA 和 GAL 的逻辑结构和工作原理及在线编程技术 ISP；第九章主要介绍了数字系统的分析与设计；第十章主要介绍硬件描述语言的基本概念及编程设计方法。

本书由魏达、高强、金玉善、曹英晖编著。其中，魏达编写了第一、二、四、六、七章，高强编写了第五章，金玉善编写了第三章，曹英晖编写了第八、九、十章。全书由魏达统稿。

由于作者水平有限，书中疏漏与不妥之处在所难免，恳请读者批评指正。

作　者

2004 年 7 月

目　录

第一章　数制与编码

本章主要介绍数字系统中数的各种基本表示方法，包括各种不同的进位计数体制及其相互间的转换；带符号数的代码表示方法；数的定点表示与浮点表示；计算机中常用的数码与字符的代码表示，如十进制数的常用编码(8421 码、2421 码、余三码)以及可靠性编码(格林码、奇偶校验码、海明码)等。

1.1　计数体制

进位计数体制即按进位方式实现计数的一种规则，是用统一的符号和规则表示数的一种方法。在日常生活中我们就是按进位计数体制来计数的，如十进制、十二进制、六十进制等等。对于任何一个数，我们可以用不同的进位体制来表示，但不同进位计数体制的运算方法和难易程度各不相同，这对数字系统的性能有很大影响。

1.1.1　十进制数

人们通常采用十进制数来计数。十进制有 10 个数字符号，即 0，1，2，3，4，5，6，7，8，9。十进制的特点是：由低位向高位的进位原则是“逢十进一”。十进制的基数为 10，基数表示某种进位体制具有的数字符号的个数。十进制的权为 10^i，它表示十进制数中不同位置上数字的固定常数。如十进制数 123.45：从左至右，第一为百位，该位置上的 1 代表 100，权为 10^2；第二位为十位，该位置上的 2 代表 20，权为 10^1；第三位为个位，该位置上的 3 代表 3，权为 10^0；小数点右边第一位为十分位，该位置上的 4 代表 4/10，权为 10^{-1}；小数点右边第二位为百分位，该位置上的 5 代表 5/100，权为 10^{-2}。

基数和权是进位体制的两个基本要素，根据基数和权的概念，我们可以将任何一个十进制数表示成多项式的形式，即一个十进制的数可以表示为

$$(N)_{10} = (N_{n-1}N_{n-2}\cdots N_1N_0N_{-1}N_{-2}\cdots N_{-m})_{10}$$

也可按权展开表示如下：

$$\begin{aligned}(N)_{10} = (&N_{n-1}\times 10^{n-1} + N_{n-2}\times 10^{n-2} + \cdots + N_1\times 10^1 + N_0\times 10^0 \\ &+ N_{-1}\times 10^{-1} + N_{-2}\times 10^{-2} + \cdots + N_{-m}\times 10^{-m})_{10}\end{aligned}$$

$$= N_{10} = \sum_{i=m}^{n-1} N_i \times 10^i, \quad 0 \leqslant N_i \leqslant 10 - 1$$

式中，n 表示整数部分的位数；m 表示小数部分的位数；10 表示基数，10^i 为第 i 位的权；N_i 表示各个数字符号。

1.1.2 二进制数

对于任意一个数，虽然原则上可以用任何一种进位体制来记数，并进行算术运算，但不同进位体制的运算方法及难易程度各不相同，因此，选择什么样的进位体制来表示数，对数字系统的性能影响很大。在数字系统中，常用二进制来表示数和进行算术运算，这是因为二进制具有如下特点：

1）二进制只有 0 和 1 两个数字符号，容易用物理状态来表示，任何具有两个不同稳定状态的元件都可以用来表示 1 位二进制数，例如：晶体管的导通和截止，脉冲的“有”和“无”等。

2）二进制运算规则简单，便于进行算术运算。其运算规则如下：

加法规则：

$0 + 0 = 0 \qquad 0 + 1 = 1$

$1 + 0 = 1 \qquad 1 + 1 = 0$（同时向相邻的高位进 1）

减法规则：

$0 - 0 = 0 \qquad 0 - 1 = 1$（同时向相邻的高位借 1）

$1 - 0 = 1 \qquad 1 - 1 = 0$

乘法规则：

$0 \times 0 = 0 \qquad 0 \times 1 = 1$

$1 \times 0 = 1 \qquad 1 \times 1 = 0$

除法规则：

$0 \div 1 = 0 \qquad 1 \div 1 = 1$

下面举例说明二进制数的四则运算。

【例 1.1】 两个二进制数相加，采用“逢二进一”的法则。

解：

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ +)\ 0\ 1\ 1\ 0 \\ \hline 1\ 0\ 0\ 1\ 1 \end{array}$$

【例 1.2】 两个二进制数相减，采用“借一当二”的法则。

解：

$$\begin{array}{r} 1\ 1\ 0\ 1 \\ -)\ 0\ 1\ 1\ 0 \\ \hline 0\ 1\ 1\ 1 \end{array}$$

【例 1.3】 两个二进制数相乘，其方法与十进制乘法运算相似，但采用二进制运算规则，对部分积进行累加时要按“逢二进一”的原则。

解：

$$
\begin{array}{rccccccc}
 & & & & 1 & 1 & 0 & 1 \\
 & & & \times) & 0 & 1 & 1 & 0 \\
\cline{4-8}
 & & & & 0 & 0 & 0 & 0 \\
 & & & 1 & 1 & 0 & 1 & \\
 & & 1 & 1 & 0 & 1 & & \\
+) & 0 & 0 & 0 & 0 & & & \\
\hline
 & 1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}
$$

【例 1.4】 两个二进制数相除，其方法与十进制除法运算相似，但采用二进制运算规则。

解：

$$
\begin{array}{r}
1010 \\
1101\,\overline{)\,10001001} \\
\underline{1101\quad\ \ } \\
010000\ \ \\
\underline{1101\ \ } \\
00111
\end{array}
$$

3）二进制还可实现逻辑运算，从而可以用布尔代数对数字系统进行分析、综合，便于逻辑电路的设计优化。

4）二进制只有两个状态，数字的传输和处理不容易出错，可靠性高。

5）采用二进制来表示数可以节省设备。

设 n 是数的位数，R 是基数，R^n 表示 n 位 R 进制数的最大信息量，则用 $n \cdot R$ 表示 R^n 个信息量所用的设备量。

例如：$n=3, R=10, R^n=10^3=1000$，则

$$n \cdot R = 3 \times 10 = 30$$

假设要表示相同或更多的信息量，即 $R^n \geqslant 1000$，取 $R=2$，则 $2^n \geqslant 1000$，由此可得 $n=10$。所要表示的信息量 $R^n=2^{10}=1024$，则设备量

$$n \cdot R = 2 \times 10 = 20$$

由此可见，对于相同的信息量，用二进制来表示数比用十进制来表示数所用的设备量要少。

由于数的表示可以使用任意进制，那么哪一种进制最节省设备呢？下面给出唯一性证明：

设 $N=R^n$，N 为需要表示的信息量，n 为 R 进制数的位数。两边取对数得

$$\ln N = n \ln R$$

令 $C=\ln N$，则

$$C = n \ln R$$

两边同乘以 R 得

$$RC = nR \ln R$$

则

$$nR = \frac{RC}{\ln R}$$

对 R 求导数得

$$0=\left(\frac{RC}{\ln R}\right)'$$

则

$$C\cdot\frac{\ln R-1}{(\ln R)^2}=0$$

所以可得

$$\ln R-1=0$$

由此得到最小的 $R=e=2.718$，则取 $R=2$。

广义地，一个 R 进制的数 N 可表示成

$$\begin{aligned}(N)_R&=(r_{n-1}r_{n-2}\cdots r_1r_0r_{-1}r_{-2}\cdots r_{-m})_R\\&=r_{n-1}R^{n-1}+r_{n-2}R^{n-2}+\cdots+r_1R^1+r_0R^0\\&\quad+r_{-1}R^{-1}+r_{-2}R^{-2}+\cdots+r_{-m}R^{-m}\\&=\sum_{i=-m}^{n-1}r_iR^i\end{aligned}$$

式中，n 表示整数的位数，m 表示小数的位数；R 为基数，在十进制中 R 应写成"10"；r_i 是 R 进制中各个数字符号，即有 $r_i\in\{0,1,2,\cdots,R-1\}$。

1.1.3 八进制数和十六进制数

由于二进制数运算规则简单，用电路实现也很方便，所以数字系统中广泛采用二进制。但当二进制数的位数很多时，书写和阅读很不方便，容易出错，记忆困难。为此，人们通常采用二进制的缩写形式八进制和十六进制。

八进制的基数 R 等于 8，每位可取 8 个不同的数字符号，即 0,1,2,3,4,5,6,7，其进位规则是"逢八进一"。由于一位八进制数的 8 个数字符号正好对应于三位二进制数的八种不同组合，所以八进制与二进制之间有简单的对应关系：

八进制	0	1	2	3	4	5	6	7
二进制	000	001	010	011	100	101	110	111

这样，八进制与二进制之间数的转换就极为方便。

【例 1.5】 将二进制数 11010.1101 转换为八进制数。

解：

$$\underbrace{011}_{3}\ \underbrace{010}_{2}.\ \underbrace{110}_{6}\ \underbrace{100}_{4}$$

所以 $(11010.1101)_2=(32.64)_8$。

由二进制转换成八进制的方法是：以小数点为界，将二进制数的整数部分从低位开始，小数部分从高位开始，每三位分成一组，头尾不足三位的补 0，然后将每组的三位二进制数转换为一位八进制数。

【例 1.6】 将八进制数 357.6 转换为二进制数。

解：

$$\begin{array}{cccc} 3 & 5 & 7. & 6 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 011 & 101 & 111. & 110 \end{array}$$

所以$(357.6)_8=(11101111.11)_2$。

十六进制的基数 R 等于 16，每位可取 16 个不同的数字符号，即 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F，其进位规则是"逢十六进一"。同理，由于一位十六进制数的 16 个数字符号正好对应于 4 位二进制数的 16 种不同的组合，所以，十六进制与二进制之间有简单的对应关系：

十六进制	0	1	2	3	4	5	6	7	8
二进制	0000	0001	0010	0011	0100	0101	0110	0111	1000
十六进制	9	A	B	C	D	E	F		
二进制	1001	1010	1011	1100	1101	1110	1111		

这样，十六进制与二进制之间数的转换也很方便。

【例 1.7】 将二进制数 1010110110.110111 转换为十六进制数。

解：

$$\underbrace{0010}_{2}\ \underbrace{1011}_{B}\ \underbrace{0110}_{6}.\ \underbrace{1101}_{D}\ \underbrace{1100}_{C}$$

所以$(1010110110.110111)_2=(2B6.DC)_{16}$。

【例 1.8】 将十六进制数 5D.6E 转换为二进制数。

解：

$$\begin{array}{ccccc} 5 & D & . & 6 & E \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 0101 & 1101 & . & 0110 & 1110 \end{array}$$

所以$(5D.6E)_{16}=(1011101.0110111)_2$。

由此可见，采用八进制和十六进制要比用二进制书写简短，易读易记，而且转换也方便，因此，计算机工作者普遍采用八进制或十六进制来书写和表达。

1.1.4 数制间的转换

由于人们习惯于使用十进制数，所以在用计算机进行信息处理时，首先必须把十进制数转换成二进制数才能被计算机所接受，然后进行运算，运算结果又必须从二进制转换成人们习惯的十进制数。任意两个 α、β 进制数之间的转换有多种方法。

1. 多项式替代法

任意两个 α、β 进制数之间的转换，其方法是：先在 α 进制中按权展开，然后替代成相应 β 进制中的数，最后在 β 进制中计算即可得 β 进制的数。

【例 1.9】 $(123.4)_8=(?)_{10}$。

解：$(123.4)_8=[1\times(10)^2+2\times(10)^1+3\times(10)^0+4\times(10)^{-1}]_8$(展开)

$=(1\times8^2+2\times8^1+3\times8^0+4\times8^{-1})_{10}$(替代)

$=(64+16+3+0.5)_{10}$(在十进制中计算)

$=(83.5)_{10}$

【例 1.10】 $(201.2)_3=(?)_2$。

解：$(201.2)_3=[2\times(10)^2+0\times(10)^1+1\times(10)^0+2\times(10)^{-1}]_3$(展开)

$=(10\times11^{10}+0\times11^1+1\times11^0+10\times11^{-1})_2$(替代)

$=(10010+1+0.101010\cdots)_2$(在二进制中计算)

$=(10011.101010\cdots)_2$

由于多项式替代法要在 β 进制中进行计算，当为十进制时，计算较方便，而当它为其他进制时，计算就很不方便。因此，这种方法用于任意进制到十进制的转换较方便。

2. 基数乘/除法

基数乘/除法分为基数乘法和基数除法两种。对于整数的转换，采用基数除法；对于小数的转换，采用基数乘法。

(1) 基数除法

【例 1.11】 将十进制整数 25 转换为二进制数，即$(25)_{10}=(?)_2$。

解：设转换结果为

$$(25)_{10}=(k_{n-1}k_{n-2}\cdots k_1k_0)_2$$
$$=(k_{n-1}2^{n-1}+k_{n-2}2^{n-2}+\cdots+k_12^1+k_02^0)_{10}$$

在十进制中计算，将上式两边除以 2，则得

$$12+\frac{1}{2}=(k_{n-1}2^{n-2}+k_{n-2}2^{n-3}+\cdots+k_12^0)+\frac{k_0}{2}$$

两数相等，则它们整数部分和小数部分必定分别相等，故有

$$12=k_{n-1}2^{n-2}+k_{n-2}2^{n-3}+\cdots+k_12^0,\quad k_0=1$$

再将上式两边同除以 2，可得

$$6=(k_{n-1}2^{n-3}+k_{n-2}2^{n-4}+\cdots+k_22^0)+\frac{k_1}{2},\quad k_1=0$$

而

$$6=(k_{n-1}2^{n-3}+k_{n-2}2^{n-4}+\cdots+k_22^0$$

可见，所要求的二进制数$(k_{n-1}k_{n-2}\cdots k_1k_0)_2$的最低位 k_0 是十进制数 25 除以 2 所得余数；次低位 k_1 是所得商 12 再除以 2 所得的余数；依次类推，继续用 2 除，直到商为 0 为止，所得的余数即为要求的二进制数 $k_0\sim k_{n-1}$之值。此方法又称除 2 取余法。

可以将上述过程写成简单算式如下：